VŠB – Technická univerzita Ostrava

Fakulta elektrotechniky a informatiky

Katedra informatiky

Software pro záznam dat měřených CCBM sondami

Software for Storing Data Measured by CCBM Probes

Rok 2013 **Roman Kaláb**

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Roman Kaláb**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: Software pro záznam dat měřených CCBM sondami
Software for Storing Data Measured by CCBM Probes

Zásady pro vypracování:

Cílem je vytvoření software pro automatizované měření CCBM sondami v prostředí Windows Mobile 6.

1. Vytvoření základního ovládacího rozhraní.
2. Ukládání dat ve formě CSV souborů.
3. Možnost nastavení parametrů měření (sériový port, čas čekání na sondu, kanály sondy).
4. Implementace algoritmu pro výpočet CRC.
5. Implementace algoritmu pro minimalizaci času měření.
6. Možnost vícenásobného měření jediným požadavkem uživatele.
7. Ošetření chyb při měření.
8. Ukládání dat do více umístění z důvodů zálohy.

Seznam doporučené odborné literatury:

Podle pokynů vedoucího bakalářské práce.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **RNDr. Lubomír Staš, CSc.**

Konzultant bakalářské práce: Ing. David Seidl

Datum zadání: 18.11.2011
Datum odevzdání: 07.05.2013

doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

PROHLÁŠENÍ STUDENTA

Prohlašuji, že jsem tuto bakalářskou/diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

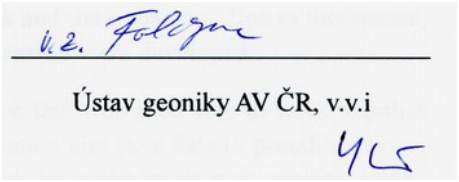
V Ostravě dne 6. května 2013

Roman Kaláb

PROHLÁŠENÍ ZÁSTUPCE SPOLUPRACUJÍCÍ PRÁVNICKÉ NEBO FYZICKÉ OSOBY

Souhlasím se zveřejněním této bakalářské/diplomové práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských/magisterských programech VŠB-TU Ostrava.

V Ostravě dne 6. května 2013



ABSTRAKT

Pro geotechniku, podzemní stavitelství a další obory je velmi důležité zjišťovat a monitorovat napěťová pole hornin a jejich změnu. Z tohoto důvodu byly vyvinuty měřící sondy, které umožňují toto napětí monitorovat.

Cílem práce je vytvoření aplikace, která bude schopna spolupracovat s těmito sondami, ověřovat správnost celé komunikace a ukládat data do přenosného zařízení. Tato aplikace musí být schopna shromáždit data ze sondy co nejrychleji, s ohledem na možnosti sondy. Aplikace také musí být co nejvíce automatizovaná. V rámci této práce byla tato aplikace kompletně vyvinuta a připravena pro nasazení.

ABSTRACT

For geotechnical engineering, underground engineering and other branches it is very important to measure and to monitor tension fields of rock and their changes. Due to this reason, monitoring probes which provide ability to monitor this tension were developed.

The aim of this bachelor thesis is to design application that would be able to work together with these probes, check correctness of whole communication and save data to portable device. This application has to be able to collect all the data from the probe as soon as possible taking into consideration abilities of the probe. Application has to be automated as much as possible. In frame of this bachelor thesis the whole application was developed and prepared for deployment.

KLÍČOVÁ SLOVA

kapesní počítač, měření tenzoru napětí, zabezpečení přenosu, automatizované měření

KEY WORDS

pocket personal computer, measurement of stress tensor, transmission securing, automated measurement

Seznam použitých symbolů a zkratek

CCBM – označení typu používané sondy (Compact Conical ended Borehole Monitoring)

GUI – Grafické uživatelské rozhraní (Graphical User Interface)

Konec řádku specifický pro systém Windows – jedná se o správnou znakovou sekvenci používanou jako ukončení řádku. Různé operační systémy používají různé znakové sekvence. Systém Windows používá kombinaci znaků CR + LF (kódy 0x0D + 0x0A). Jiné systémy mohou využívat jiné znakové sekvence (např. Unix a Unix-like systémy jako Linux, AIX, OS X a Oracle Solaris využívají pouze znak LF – 0x0A, RISC OS využívá opačnou kombinaci, tedy LF + CR – 0x0A + 0x0D). Bude-li kdekoliv dále v textu zmíněn konec řádku, je myšlen konec řádku typický pro systém Windows, není-li řečeno jinak. (Informace převzaty z *[1]*.)

OS – Operační systém

PDA – Personal Digital Asistent (Osobní digitální asistent), dnes obvykle označován jako kapesní počítač

ÚGN – Ústav geoniky Akademie věd ČR, v.v.i.

Obsah

1. Úvod

1.1. Motivace

Ústav geoniky AV ČR, v.v.i se již několik let zabývá vývojem měřících napěťových sond. Tyto sondy prošly od svého vzniku značným vývojem, díky čemuž vznikl požadavek i pro obnovení platformy pro ovládání sond a sběr dat.

Původní řešení této problematiky bylo implementováno za pomoci mobilního systémy Psion Workabout. Tento systém je v dnešní době již zcela nedostatečný, a to hned z několika důvodů.

Především novější generace sond využívají adresování a z důvodů zabezpečení korektní komunikace i CRC16 kontrolní součty. Používaný Psion Workabout však měl příliš nízký výkon (konkrétně procesor taktován na frekvenci 27MHz) a příliš málo místa v paměti, než aby bylo reálné provádět výpočet CRC součtů v průběhu práce se sondami. Z tohoto důvodu byly všechny zprávy odesílané sondě předprogramovány v tomto systému, včetně kontrolních součtů. Díky tomu nebylo možné využívat adresaci sond, veškerá komunikace musela probíhat přes příkazy adresující všechny jednotky (viz kapitola 4.2). Kontrolní součty v příchozích zprávách byly ignorovány. Z tohoto důvodu vyvinuté zabezpečení pomocí kontrolních součtů naprosto postrádalo smysl.

Dalším problémem systému Psion Workabout byla velmi malá paměť. Tento systém disponuje pouze 2MB RAM pamětí, do které se ukládají jak data, tak veškeré programové vybavení. Systém lze rozšířit o dvě paměťové karty ve velikosti maximálně 8MB na každou, tedy do maxima 16+2MB, avšak tyto karty jsou v dnešní době již prakticky nedostupné. I v případě možnosti sehnat tyto karty je však dostupný prostor stále malý pro požadované využití.

Vzhledem k tomu, že systém nevlastní žádnou trvalou paměť, dochází také ke ztrátě dat a instalovaného software při každé výměně baterií či při jejich vybití. Měření prováděná v dolech není možno žádným způsobem opakovat, tedy při každé takovéto situaci dojde k nenávratné ztrátě informací o dané lokalitě.

Již poměrně nepatrnými nevýhodami se jeví nemožnost soubory ze systému Psion Workabout synchronizovat s PC a nemožnost spouštět instalovaný software automaticky po startu systému. Všechny operace s naměřenými daty musely být prováděny ručně. V průběhu tohoto procesu docházelo často k uživatelským chybám a dalším ztrátám dat.

Ze všeho výše uvedeného jasně vyplývá, že při dnešním vývoji technologií kapesních počítačů je toto řešení již zcela zastaralé a prakticky nepoužitelné. Byla zvažována možnost nákupu nového systému Psion. Některé nedostatky z původní verze by se sice odstranily (například příliš nízký výkon), některé by však vzhledem k nezměněnému základu OS přetrvaly i nadále. Dalším argumentem také byla cena, protože vybrané odolné PDA (viz kapitola 2.2) je podstatně levnější než nový systém Psion, a to i přesto, že ho v mnoha ohledech předčí (např. výkon, dostupná paměť a v neposlední řadě také odolnost proti pádům, vodě a prachu).

Obr. 1 – Standardní sonda CCBM 2. generace s přenosným terminálem Psion Workabout

1.2. Zadání práce

Předmětem práce je vytvořit aplikaci, která bude pokud možno odstraňovat nedostatky původního řešení pomocí sytému Psion Workabout. Dále by měla umožňovat velmi jednoduché ovládání a časově efektivní měření napěťových sond, které bude využívat všech komunikačních funkcí sondy (především výpočty kontrolních součtů CRC16 a jejich ověřování a přímé adresování sond). Zároveň také tato aplikace musí obsahovat rozhraní, které by umožňovalo kompletní nastavení.

Aplikace musí být schopna ukládat všechna data z komunikace, rozdělená podle jednotlivých sond. Dále je potřeba, aby bylo možno provést několik měření těsně po sobě při jediném požadavku uživatele.

Aplikace by měla zobrazovat průběh celého měření, aby bylo zřejmé, že měření probíhá a nedošlo k žádné chybě. Také bude rozpoznatelný přibližný čas zbývající do konce měření.

1.3. Cíl práce

Cílem práce bylo vytvořit aplikaci CCBM Reader, která by pokud možno kompletně splňovala zadání, s tím, že hlavní důraz je kladen na co nejvyšší stupeň automatizace a jednoduchost ovládání.

Dalším cílem bylo vytvořit aplikaci, která by šla poměrně jednoduše rozšířit pro použití s jiným typem sond (při zachování stejného typu komunikace) či o základní výpočty nad měřenými daty (například o průměrování naměřených hodnot).

V práci nejde o implementaci všech funkcí podporovaných sondou. Sondu lze za pomoci komunikačního protokolu (popsáného dále v této práci v kapitole 4) i nastavovat – například změnit její číslo, název, adresu, kalibrovat ji a měnit její další nastavení.

Aplikace v této verzi také nemá mít žádné schopnosti zpracování naměřených dat, s výjimkou převodu naměřených hodnot z formátu zasílaného sondou do čísel v zadaném rozsahu. Zpracovávání dat se bude realizovat až na klasickém počítači, po kompletaci všech dat ze všech sond a PDA.

1.4. Volba platformy a jazyka

Počátečním rozhodnutím samozřejmě bylo, o jaké zařízení by se mělo jednat. Nakonec byl zvolen kapesní počítač. Musel být vybrán takový, který by splňoval všechny požadavky pro práci v určeném nasazení.

V oblasti platformy nebyla možnost výběru. Z důvodů, které budou uvedeny v příští kapitole, nebyla jiná varianta, než Windows Mobile 6. Jediná možnost výběru byla mezi verzemi – a to Windows Mobile 6 nebo Windows Mobile 6.1. Vzhledem k tomu, že verze 6.1 obsahuje velké množství oprav, vyladění výkonu a aktualizaci některých základních aplikací, byla zvolena tato novější verze.

Na základě tohoto rozhodnutí byla velmi zúžena možnost výběru programové platformy a programovacího jazyka. Na Windows Mobile jsou podporovány pouze nativní aplikace, aplikace fungující pod rozhraním .NET Compact nebo Java aplikace.

Java aplikace je ale možné spouštět pouze v emulátoru, který obsahuje jen málo knihoven a ve verzi pro Windows Mobile má bohužel velké množství chyb a nedostatků. Také neumožňuje přístup k hardware, takže tato varianta byla hned na začátku vyřazena.

Nativní aplikace sice dosahují o něco vyššího výkonu, nicméně v .NET Compact frameworku existuje velké množství již připravených funkcí, které tak není potřeba znova vyvíjet. Také tvorba grafického rozhraní je značně jednodušší. Tyto předpřipravené funkce jsou také velmi dobře optimalizovány, díky čemuž reálné rozdíly ve výkonu jsou minimální nebo dokonce žádné.

Co se týče výběru programovacího jazyka v .NET Compact frameworku, v podstatě není důležité, který je zvolen (viz kapitola 3.3). Já jsem si zvolil jazyk C#.

1.5. Použité vývojové nástroje

K vývoji aplikace jsem použil následující nástroje:

- NOTEPAD++ V6.3.2 (UNICODE) – Textový editor, použit pro psaní kódu a základní syntaktickou kontrolu (dostupný z <http://notepad-plus-plus.org/download/>)
- MICROSOFT DEVICE EMULATOR V3 – Emulátor Windows Mobile 6.0, testování aplikace (dostupný z <http://download.microsoft.com>)
- WINDOWS MOBILE 6.1.4 EMULATOR IMAGES – Balíček s obrazy systémů pro Device Emulator (dostupný z <http://download.microsoft.com>)
- CSC.EXE – kompilátor kódu pro .NET Framewrok (součást .NET Framework v3.5, dostupný z <http://download.microsoft.com>)
- CABWIZ.EXE – Program pro tvorbu instalačních balíčků pro Windows Mobile (součást Windows Mobile 6 Professional and Standard Software Development Kits Refresh, dostupný z <http://download.microsoft.com>)

2. Hardware + použití

2.1. Informace o použití

Celá aplikace je určena pro použití v dolech. Vše, ať už se jedná o hardware nebo software, musí umožňovat velmi jednoduché ovládání. Za běžných podmínek bude celou aplikaci po úvodním nastavení obsluhovat už pouze školený personál dolu. Z tohoto důvodu není aplikace uzpůsobena k pravidelnému nastavování. Naopak uživatelské rozhraní je velice jednoduché, aby nedocházelo ke zbytečným chybám uživatele jako je nesprávný postup měření, nechtěné či špatné přenastavení měření a podobně.

2.2. PDA

Jak jsem již uvedl, bylo rozhodnuto pro celou aplikaci použít kapesní počítač (PDA). Vzhledem k uvedenému využití v dolech musel také splňovat poměrně náročné požadavky. Velmi důležitým parametrem byla vysoká odolnost proti nárazům a otřesům. Dále byla potřeba odolnost proti vodě a prachu. Bylo také nutné, aby PDA bylo vybaveno sériovým portem.

Na trhu existuje jen málo výrobců vyrábějících zařízení splňující tyto požadavky. Nakonec byly vybrány počítače firmy Nautiz.

Tato firma vyrábí 5 základních modelů odolných PDA. Jedná se o modely eTICKET PRO, X1, X3, X5 a X7.

eTICKET PRO, jak název napovídá, je navržen především pro použití ve veřejné dopravě a pro provádění a ověřování platebních transakcí. Toto PDA je vybaveno čtečkou RFID tagů, fotoaparátem a 2-D čtečkou kódů, Bluetooth, WiFi, volitelně GSM modulem a také má dostupný sériový port. Nicméně sériový port poskytuje pouze TTL úrovně, což neodpovídá RS – 232 standardům. Také není vhodné pro toto použití, protože krytí proti vodě a prachu má pouze IP65.

U modelu X1 se jedná spíše o odolný mobil, sice velmi vybavený, nicméně díky své konstrukci bude podstatně méně odolný, než ostatní PDA vyráběná firmou Nautiz. Také není vybaven sériovým portem.

Model X3 už patří mezi odolnější PDA, nicméně opět není vybaven sériovým portem. Navíc splňuje pouze krytí IP65.

Model X5 byl jedním z kandidátů na použití. Je vybaven Bluetooth, WiFi, infračerveným portem, ethernet portem (použitelným při připojení na stolní kolíbku), volitelně GSM modulem a sériovým portem. Nakonec také nebyl použit, protože sériový port pracuje pouze na TTL úrovních a krytí je jen IP65.

Model X7 se stal modelem, který bude na celou aplikaci použit. Díky své konstrukci vypadá i nejodolněji ze všech PDA – má poměrně dost robustní gumové spodní rohy a celou horní část

(zakrývající slot na SD kartu), display je asi 7mm pod úrovní krytu (měřeno proti nejvyššímu bodu gumového krytu).

PDA je vybaveno Bluetooth, WiFi, volitelně GSM modulem (nebude do PDA osazován), 4GB vnitřní paměti (iNAND flash), čtečkou SD karet, sériovým portem, GPS, 3 megapixelovým fotoaparátem, a volitelně také skenerem kódů a čtečkou RFID (také nebudou osazovány).

Tento model má velkou výhodu v tom, že sériový port, jímž je vybaven, splňuje specifikace RS – 232. Také 4GB paměť, která se jinde neobjevuje, bude použita jako záložní paměť pro ukládání dat měření. Hlavním úložištěm by měla být paměťová karta SD.

Také krytí proti vodě a prachu splňuje vyšší používanou specifikaci IP 67. Toto PDA může být používáno v teplotách od -30°C do 60°C a v relativní vlhkosti až 90%. Dle specifikací vydrží 26 pádů z 1,22m, 6 dalších pádů při teplotě -30°C a 6 dalších při 60°C. Díky tomu by mělo být pro určené využití opravdu vhodné.

V PDA je osazen procesor Marvell PXA310 o rychlosti 806 MHz a SDRAM paměť s kapacitou 128 MB, což je pro běh vyvíjené aplikace naprosto postačující.

Obr. 2 – PDA, které bude využito pro provoz aplikace – Nautiz X7

Obr. 3 – PDA dovybaveno o napájecí modul (ve spodní části) a kožený obal

2.3. Sonda

2.3.1. Základní informace o sondě a jejím vývoji

Napěťová pole jsou jedním ze základních faktorů, které spolu s mechanickými a přetvárnými vlastnostmi hornin, rozhodným způsobem ovlivňují chování horského masivu. Proto má zjišťování, popř. monitorování, změn napěťových polí v horském masivu zásadní význam pro geotechniku, podzemní stavitelství a další obory zabývající se napěťovými stavy a projevy napětí v zemské kůře. Data získaná z výzkumu velikosti, směrů a změn napětí v horském masivu mají zásadní přínos pro oblast matematického modelování, kde mohou být použita jako vstupní parametry modelu napěťových polí za účelem získání přesnějšího obrazu o chování horského masivu a podzemních objektů v předmětných oblastech.

Ústav geoniky AV ČR, v.v.i se již řadu let zabývá problematikou napěťového stavu masivu in-situ, zejména v oblasti české části hornoslezské pánve. Původně se provádělo měření napětí měřícím zařízením na bázi hydraulického štěpení stěn vrtu. Vzhledem k tomu, že zařízení má ale i značná omezení a ne všude je použitelné, bylo vyvinuto zařízení, které by umožňovalo odstranit nebo alespoň minimalizovat nedostatky původního zařízení.

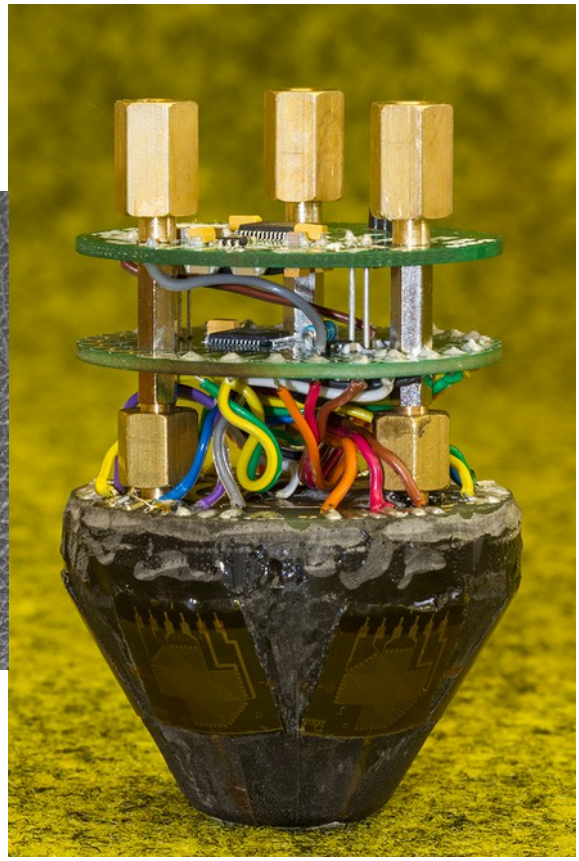
Již od roku 2003 vyvíjí ÚGN tenzometrické sondy, původně založené na japonské metodice CCBO (Compact Conical ended Borehole Overcoring) vypracované prof. K. Sugawarou, prof. Y. Obarou a S. S.Kangem z Kumamoto University *[2] (Obr. 4)*. Od té doby prošly sondy řadou změn. Jedná se o změny ve způsobu měření (jiné tenzometry), v napájení (původně integrované napájecí články v sondě, nyní externí napájení) i v přenosu dat. Původní přenos dat probíhal přes infračervený přenos, komunikace byla nezabezpečená. Později byla vyvinuta sonda CCBM druhé generace, která již využívala sériovou komunikaci po kabelu. Aktuální sondy – tedy CCBM 2,5. generace – již využívají sériový přenos dat se zabezpečením pomocí kontrolních součtů. Vzhledově se sonda 2. a 2,5. generace prakticky neliší (kromě jiných rozměrů tenzometrů). Na fotce *(Obr. 1)* je CCBM sonda 2. generace s terminálem Psion Workabout, který byl původně používán pro záznam dat. (Informace převzaty z *[3]*.)

2.3.2. Technické parametry standardní sondy CCBM 2. generace

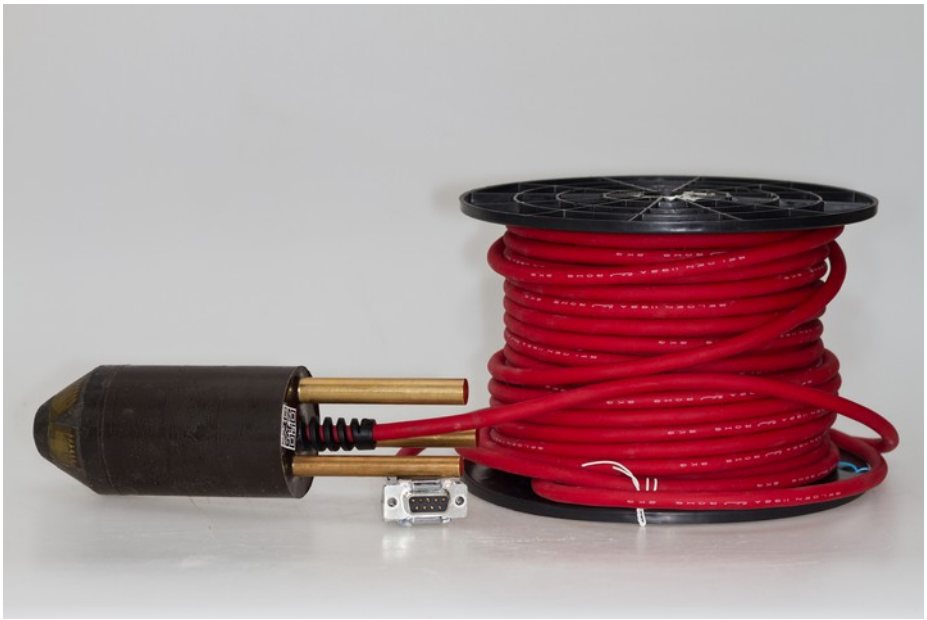
průměr sondy	=	50 mm ± 2 mm
snímač	=	6 dvojic tenzometrických mřížek rozet s dvěma mřížkami o velikosti 3x3 mm se a jmenovitým odporem 350 Ω
průměr vrtu	=	76 mm
úhel kužele	=	60° (vrcholový úhel)
elektrický šum	≤	± 60 μstrain (v prostředí bez EMI, se stand. rozsahem ± 16 000 μstrain)
rozlišení	=	cca 2 LSB / μstrain (ve standardním rozsahu)
hmotnost	=	cca 230 g (bez kabelu)
příkon	≤	1 W
datový výstup		dle standardu TIA-232-F
formát dat		proprietární protokol založený na Tedia AIBus 2 (viz *[4]*)

Technické údaje o sondě byly převzaty z *[3]*.

Obr. 4 – Japonská měřicí sonda CCBO



Obr. 5 – Sonda CCBM 2,5. generace, pohled dovnitř



Obr. 6 – Hotová sonda CCBM 2,5. generace s kabelem, připravena pro umístění do vrtu

3. Windows Mobile + .NET Compact 3.5

3.1. Windows Mobile 6.1 Classic

Windows Mobile ve verzi 6 a podverzích nabízí 3 různé edice. Každá má své určení a specifika. V každé jsou také obsaženy jiné aplikace vzhledem k jejímu určení.

Edice Professional je určena pro všechna zařízení, která obsahují GSM nebo CDMA modul a dotykovou obrazovku (v originále firmou Microsoft označována jako „Pocket PC – with Mobile Phone“). Jedná se o nejvybavenější edici obsahující všechny ovladače a aplikace.

Edice Standard je určena zařízením s GSM nebo CDMA modulem bez dotykové obrazovky (Microsoft označení „Smartphone – without Touch Screen“). Na základě toho samozřejmě neobsahuje ovladače pro dotykovou obrazovku, zato však jako jediná edice obsahuje upravenou hlavní obrazovku, která byla aktualizována ve Windows Mobile 6.1 oproti verzi 6. Tato edice také na rozdíl od ostatních dvou neobsahuje aplikaci pro vzdálený přístup k PC (Remote Desktop Mobile) a má omezené funkce v kancelářském balíku Office Mobile.

Edice Classic je pro zařízení s dotykovou obrazovkou bez GSM nebo CDMA modulu (Microsoft označení „Pocket PC – without Mobile Phone“). Proto jí chybí ovladače pro tento hardware a software pro nastavení přístupu k internetu přes mobilní technologie. Také chybí software pro IP telefonování.

V kapesním počítači se nachází výše zmíněný systém Windows Mobile 6.1, vzhledem k hardwarovému vybavení konkrétně v edici Classic.

3.2. Instalace software v prostředí Windows Mobile

Veškerý software v prostředí Windows Mobile může být kopírován do zařízení a spouštěn v něm přímo. Tato metoda je ale složitější a nenabízí ani zdaleka tolik komfortu jako instalace. Instalační soubory pro Windows Mobile mají příponu CAB, mají ale svůj specifický formát (nejedná se o klasické CAB archivy ani o CAB archivy používané pro Windows instalátory). Tento formát je shodný s instalačním souborem pro Windows CE.

V těchto instalačních archivech jsou jednak komprimovány všechny soubory, zároveň je ale v archivu přiložen soubor s příponou 000. V tomto souboru je uloženo množství dalších informací. Jedná se například o základní informace jako minimální a maximální podporovaná verze OS, jméno aplikace, jméno společnosti, ale také informace o nepodporovaných platformách, podporovaných typech CPU a podobně.

Dále je v archivu přibalen soubor *_setup.xml*. Tento soubor obsahuje další informace používané pro instalaci. Některé z těchto informací mohou být umístěny v souboru 000, místo v tomto souboru, ale u novějších instalačních archivů se tato možnost nevyužívá. Některé informace jsou vkládány duplicitně do obou souborů. Tento soubor obsahuje především informace o operacích se soubory – tedy jejich původní názvy (soubory jsou při tvorbě archivu

přejmenovávány do formátu názvu 8+3 znaky, kde přípona je pořadové číslo souboru) a cílové adresáře pro instalaci.

Soubor dále obsahuje také kompletní informace o úpravách v registrech a o tvorbě zástupců. Může také obsahovat informaci o instalační knihovně, která se spouští místo standardní instalační knihovny Windows Mobile nebo po ukončení standardní instalace a další nastavení. Není nutné, aby obsahoval všechny tyto sekce, pokud nejsou při instalaci využity.

Celý instalační archiv je standardně tvořen programem CabWiz (viz kapitola 1.5), ačkoliv je ho možné pochopitelně vytvořit i ručně. Vzhledem k uvedeným dvěma souborům je však tento postup velmi zdlouhavý a náročný, protože všechny informace je nutné ručně převést do binárního formátu (viz *[5]*) a uložit je do souboru 000 (nejjednodušeji asi hexadecimálním editorem) a zbylé informace převést do XML formátu.

Při tvorbě archivu programem CabWiz je potřeba nejprve ručně napsat konfigurační soubor (s příponou INF), který obsahuje všechny informace potřebné pro tvorbu archivu, souboru 000 a souboru *_setup.xml*. Patří sem prakticky vše, co se může objevit v souboru 000 nebo zmíněném *_setup.xml* a některé další informace. Kompletní popis všech nastavení možných v tomto souboru lze nalézt v *[6]*.

Kompletní návod pro ovládání programu CabWiz je uveden na stránkách Microsoftu *[7]*.

3.3. Framework .NET Compact 3.5

Jedná se o aplikační rozhraní vyvinuté firmou Microsoft. Toto rozhraní poskytuje mezivrstvu mezi OS a aplikací. Prostředí .NET Frameworku nabízí jak spouštěcí rozhraní, tak i všechny potřebné knihovny pro spuštění aplikací. Proto jsou aplikace vytvářené na této platformě poměrně malé. Díky tomu, že .NET Framework nabízí pro velké množství používaných úkonů již připravené funkce, programátor má ulehčenou práci, protože je nemusí znova vytvářet. Tyto funkce jsou také velmi dobře optimalizovány pro danou platformu, na které jsou spouštěny, a proto jsou obvykle rychlejší nebo alespoň stejně rychlé jako ty, které by bylo možné vytvořit ručně.

Platforma .NET jako taková nevyžaduje použití žádného konkrétního jazyka. Všechny jazyky se překládají do mezijazyka Common Intermediate Language (tzv. bytecode). Aktuálně nejpoužívanější jsou jazyky Visual Basic .NET, C# a C++, ale existuje i mnoho dalších, například F#, J#. Některé ani nejsou vyvíjeny firmou Microsoft, kupříkladu Delphi, IronPython a další.

Existuje poměrně mnoho rozdílů mezi klasickou verzí .NET Frameworku a verzí pro mobilní telefony a kapesní počítače, tedy .NET Compact Frameworku. Především se jedná o mnohem menší počet implementovaných funkcí a ovládacích prvků použitelných pro formuláře.

Bohužel omezený počet implementovaných funkcí je věc, na kterou programátor naráží poměrně často. Tyto funkce je pak potřeba nahrazovat za použití více jiných funkcí a výsledek mnohdy bývá mnohem pomalejší než funkce dostupná v klasickém frameworku.

Dalším problémem je, že .NET Compact Framework již není nijak aktualizován. Poslední a nejaktuálnější verze je 3.5, která byla vydána v roce 2007. Do novější verze .NET Framework 4 však přibylo poměrně hodně funkcí, které ještě zvyšují rychlost aplikace. Například se jedná o funkci „IsNullOrWhiteSpace“ (funkce, která zjistí, zda je proměnná řetězce nastavena na hodnotu null, řetězec je prázdný nebo obsahuje pouze tzv. whitespace znaky). Ve verzi 3.5 je nutné tuto funkci složit ze dvou různých funkcí – a to z funkce „Trim“ (oříznutí řetězce o všechny prázdné znaky zleva i zprava) a funkce „IsNullOrEmpty“ (zjištění, zda je proměnná řetězce nastavena na hodnotu null nebo je řetězec prázdný). Takto složená funkce je však zhruba 11x pomalejší než funkce využitelná v novější verzi.

Společnost Microsoft poskytuje pro vývoj aplikací v .NET Frameworku vývojové studio Microsoft Visual Studio. Konkrétně pro vývoj aplikací pod .NET Compact Framework je uzpůsobeno Visual Studio 2003, 2005 a 2008. V novější verzi Visual Studio 2010 již byla možnost využití Compact Frameworku nahrazena novějším produktem Silverlight, který je využíván pro vývoj aplikací pro Windows Phone 7.

Vzhledem k tomu, že všechny edice Microsoft Visual Studia jsou buď placené nebo je není možné použít pro komerční účely, rozhodl jsem se použít sadu aplikací zmíněnou v kapitole 1.5.

Informace o .NET Compact Frameworku 3.5 a některé programovací techniky byly převzaty z *[8]*.

4. Komunikační protokol sondy (CCBM)

Komunikační protokol sondy je proprietární – založený na AIBus-2 od fy Tedia.

Základní verze protokolu je k dispozici v elektronické podobě na stránkách firmy.

Komunikační protokol AIBus-2 pro svoji topologii „master – slave“ využívá adresace koncových zařízení (tzn. „slave“ jednotek) prostřednictvím 8-bitové adresy; na síti tedy může kromě řídícího systému („masteru“) současně spolupracovat maximálně 256 jednotek. Praktická realizace však tento počet omezuje o dvě vyhrazené (0 a 255) na konečných 254.

Adresa „0“ je určena pro úvodní inicializaci jednotky, adresa 255 pak umožňuje předávání společných příkazů současně všem jednotkám bez ohledu na jejich adresu. (Citováno z *[4]*.)

Komunikační protokol není nijak závislý na hardwarové implementaci, nejobvyklejší je podle fy Tedia rozhraní RS - 485. Hardwarové rozhraní v použitých sondách je však ve standardu RS - 232, konkrétně s rychlostí 9600 baud, 8 datových bitů, 1 stop bit a sudá parita.

4.1. Stavba příkazů

Příkazy jsou stavěny po jednotlivých bytech podle následující tabulky.

adr	fn	ch	cmd	d0	d1	d2	d3	CRC16	CRC16

Příklad příkazů na přečtení prvních dvou kanálů sondy s adresou 7:

PC>	07	00	00	00	00	00	00	00	01	ED	(HEX)
CCBO<	07	00	00	01	EB	7F	00	00	38	11	(HEX)
PC>	07	00	01	00	00	00	00	00	00	3C	(HEX)
CCBO<	07	00	01	01	EA	7F	80	00	xx	xx	(HEX)

Hodnota ch0: 32 747

Hodnota ch1: -32 747

adr – adresa sondy na sběrnici
fn – požadovaná funkce (0 = měření na daném kanálu)
ch – číslo kanálu v sondě
cmd – informační byte
d0 – nižší byte čísla
d1 – vyšší byte čísla
d2 – znaménko (0 nebo 0x80)
d3 – vždy 0 (nepoužito)
2B – CRC16 (výpočet CRC viz popis Tedia AIBus2 *[4]*)

4.2. Adresování sond

Sondy se standardně adresují přímo v příkazu, jak je vidět v předchozí kapitole.

V protokolu lze také využít implementovanou funkci adresování všech zařízení na sběrnici (tzv. broadcast s adresou 0xFF neboli 255). Vzhledem k tomu, že v protokolu není nijak implementována funkce pro sledování stavu přenosového média ani pro detekci kolize, není ve většině případů, kdy mají zařízení odpovídat na příkaz, vhodné tuto možnost využívat. Za předpokladu, že by na sběrnici bylo připojeno více než jedno zařízení, začala by všechna zařízení vysílat prakticky současně (s nepřesností několika milisekund díky rozdílným délkám měření) a data by byla nerozpoznatelná.

V aplikaci je přesto funkce adresování všech zařízení použita, protože aktuálně využívané sondy se připojují přímo ke kapesnímu počítači, a tedy nehrozí připojení více zařízení najednou. Díky tomu si mohu dovolit tuto funkci protokolu použít a velmi tak zrychlit nalezení adresy připojené sondy. Sonda totiž za všech okolností odpovídá svou adresou, a to i v případě, že byla adresována všechna zařízení na sběrnici. Za předpokladu, že by nebylo možné tuto funkci využít, značně by se prodloužil čas potřebný pro nalezení sondy, protože jediná jiná varianta, jak nalézt adresu sondy připojené na sběrnici, je procházet všechny dostupné adresy po jedné. To představuje v nejhorším případě až 254 pokusů (vzhledem k tomu, že celkem lze adresovat až 254 zařízení).

4.3. Příkazy používané při komunikaci se sondou

Měření kanálů je samozřejmě základní funkcí sondy. Přesný popis komunikace pro měření je uveden výše (viz kapitola 4.1).

Po komunikačním protokolu lze sondě poslat mnohem více příkazů. Jediná další využitá funkce sondy v aplikaci je funkce pro načtení informací ze sondy (funkce s kódem 0x02).

Tato funkce umožňuje načíst téměř všechny nastavitelné informace o sondě. Odpovědí na tento požadavek je 5 po sobě následujících zpráv. Pro aplikaci je podstatná pouze první zpráva, ve které je uvedena adresa sondy (vzhledem ke specifikaci protokolu), číslo sondy a její název. V tabulce lze vidět obecný formát zprávy (první řádek), konkrétní informace (druhý řádek) a vzorovou zprávu (třetí řádek).

adr	fn	ch	cmd	d0	d1	d2	d3	CRC16	CRC16
adr	fn	ch	cmd	**name_1**	**name_2**	**name_3**	**num**	CRC16	CRC16
0x07	0x00	0x00	0x01	0x55	0x47	0x4E	0x07	0xE9	0x96

Jak je vidět výše, název může obsahovat maximálně 3 znaky (*name_1* až *name_3*). Celý název je pak sestaven v pořadí, v jakém se znaky ve zprávě nacházejí. Jedná se konkrétně o číselné vyjádření znaků podle tabulky ASCII (v tomto případě konkrétně 0x55 – U, 0x47 – G, 0x4E – N, tedy název sondy je UGN). Z tohoto je zřejmé, že teoreticky lze využít i netisknutelné, řídící a speciální znaky či znaky s diakritikou. V praxi tato možnost však využita není a není ani doporučena, protože název sondy je využíván ke tvorbě názvů datových

souborů a především k jednoznačné identifikaci sondy, což by mohlo působit problémy při různých místních nastaveních systému (znaky v horní polovině ASCII tabulky jsou závislé na zvolené znakové sadě, mohlo by tedy docházet k záměně znaků na různých systémech).

5. Analýza řešení

5.1. Návrh

Při návrhu celé aplikace musel být brán v potaz požadavek na co nejjednodušší ovládání, které minimalizuje možnost uživatelských chyb a nesprávného nastavení a zároveň požadavek na implementaci pokročilého konfiguračního rozhraní. Tyto poměrně protichůdné požadavky musejí být sloučeny v jedné aplikaci. Z hlediska GUI a jeho přehlednosti také narážíme na problém značně malého display, a tedy malého počtu prvků umístitelného na obrazovku najednou.

Dále musel být zohledněn nízký výpočetní výkon, nízká rychlost paměti a paměťových médií a malá paměť RAM. Celá aplikace proto musela být navržena tak, aby pokud možno spotřebovávala co nejméně systémových prostředků.

5.2. Technické problémy implementace

K jednomu z nejzávažnějších problémů, které se objevily při běhu aplikace, patřila problémovost komunikace se sondou. Tato komunikace je realizována za použití standardu RS - 232, konkrétně za použití dvouvodičové komunikace. Tento typ komunikace nemá žádné řízení toku dat, není tedy možné nijak sledovat stav sondy a její připravenost k vyslání dat. Stejně tak sonda není schopna určit, kdy má data začít vysílat a přenos dat tak začíná ihned po naměření požadované hodnoty. Tato doba je však značně proměnlivá, u korektního měření se pohybuje zhruba v rozmezí 10 ms až 120 ms. Dalším nedostatkem v této oblasti je, že sonda někdy odesílá data se značným zpožděním, a to až 0,7 s. Někdy se také stává, že data neodešle vůbec.

Dalším problémem byly GUI komponenty obsažené v .NET Compact Framework. Konkrétně jsem narazil na dva případy, a to dialog pro výběr souboru, který sice obsažen je, ale neumožňuje žádným jednoduchým způsobem výběr složky. OS sám prochází celý souborový systém a zobrazuje v jednom okně najedou všechny soubory konkrétní přípony, která je nastavena. Tento dialog může být nahrazen klasickým dialogem pro výběr souborů, jak je uživatel zvyklý z běžných Windows, avšak pouze za pomoci programů třetích stran (tuto možnost podporuje např. program SPB Pocket Plus). Další problematickou komponentou je výběr složky, protože toto je komponenta, která je obsažena pouze v klasických Windows a jejich rozhraní, ve Windows Mobile nikoliv. Vzhledem k výše uvedenému jsem se rozhodl pro oba dialogy implementovat vlastní formuláře, které by umožnily jednoduchý a přehledný výběr složky nebo souboru.

5.3. Zvolené postupy

Z pohledu GUI jsem se rozhodl pro rozdělení celé aplikace na dva uživatelské formuláře. Jeden hlavní, ve kterém bude zadáván požadavek na změření sondy. Budou se v něm zobrazovat všechny informace a průběh celého měření. Tento požadavek bude zadáván za pomoci jednoho poměrně velkého tlačítka. Jeho velikost se může jevit přehnaná, ale při jeho návrhu byli bráni v potaz jako potencionální uživatelé především horníci, kteří budou PDA obsluhovat bez použití stylusu, pouze prsty a ve značném množství případů pravděpodobně i v rukavicích. Z tohoto důvodu není ani v blízkosti tohoto tlačítka žádný jiný ovládací prvek. Tím je do značné míry zabezpečeno, že nebude docházet k „přehmatům“ a nechtěným aktivacím jiných ovládacích prvků.

Ve druhém formuláři bude umožněna konfigurace celé aplikace. Tento formulář bude dostupný přes položku v místním menu programu. Ovládací prvky v tomto formuláři jsou optimalizovány pro použití stylusu. Každé PDA je jím vybaveno a stylus je uložen v krytu PDA na zadní straně, neměl by tedy být problém tento požadavek splnit. I přes poměrně malé ovládací prvky však nebylo možné všechna nastavení umístit na jednu obrazovku. Rozhodl jsem se tedy využít ovládacího prvku „kartotéka“, který umožňuje využít několik prakticky celoobrazovkových karet (pouze zmenšených o řádek s názvy). Tímto způsobem jsem dosáhl přehledného, přívětivého a dobře ovladatelného rozhraní.

V aplikaci také musela být ošetřena komunikace se sondou, s ohledem na uvedenou nespolehlivost a značný rozptyl doby odpovědí. Vzhledem k tomu, že se jedná o poměrně rozsáhlé téma, rozhodl jsem se mu věnovat samostatné kapitoly 6.1 a 6.2.

Aplikace má být zrealizována tak, aby se zobrazoval průběh měření. Požadavek byl směřován spíše pro orientační účely, ne pro nějaké konkrétní časové výpočty. Z tohoto důvodu jsem se rozhodl umístit na hlavní formulář ovládací prvek „ukazatel průběhu“ (progressbar). Bylo několik možností, jak tento prvek uvést do provozu. Nakonec jsem se rozhodl pro variantu komunikace částí aplikace pomocí zpráv. Hlavní formulář tedy spustí modul pro měření a začne odchytávat zprávy o průběhu měření. Měřící modul pak tyto zprávy generuje v průběhu měření. Bylo potřeba zvolit vyvážení mezi vysokým zatížením procesoru a pomalou a skokovou aktualizací ukazatele průběhu. Jako optimální se jeví aktualizovat informaci o průběhu s každým jednotlivým měřením kanálu. Bude tedy generován počet zpráv odpovídající součinu počtu kanálů a počtu měření na jeden požadavek uživatele. Dalo by se dosáhnout ještě vyššího rozlišení průběhu, a to použitím informace o maximálním počtu pokusů na jednotlivých kanálech, ale zatížení generované tímto postupem by již bylo příliš vysoké a značně by zpomalilo průběh samotného měření.

Aplikace má aktuálně za úkol provádět se sondou v podstatě pouze dvě operace, a to načtení informací o sondě a měření na uživatelem definovaných kanálech. O obsluhu těchto operací se stará modul měření, který má pro každou z nich implementovánu samostatnou funkci. Kromě těchto dvou má ještě implementovánu funkci pro výpočet CRC kontrolního součtu zpráv.

Vzhledem k tomu, že PDA by nemělo být využíváno k jiným účelům než k realizaci měření, rozhodl jsem se po konzultaci se zadavatelem startovat aplikaci automaticky po startu systému. Tohoto je docíleno umístěním zástupce aplikace do složky automatického spouštění (tzv. StartUp). Tento zástupce je automaticky vytvořen při instalaci aplikace. Aplikace také byla nastavena tak, aby se maximalizovala na celou obrazovku. Není tedy možné standardním způsobem využít nabídku Start. Stále je možné dostat se do nabídky Start klávesou na klávesnici, ale toto není možné vzhledem k hardwarovým omezením nijak změnit. Ve Windows Mobile je totiž možnost v programu namapovat pouze některé hardwarové klávesy, výchozí klávesy (např. Windows Logo a OK) však přemapovat není možné. I přesto se tímto způsobem dosáhlo určitého „zabezpečení“ a zjednodušení, zvláště pro personál dolů.

5.4. Grafický návrh GUI aplikace

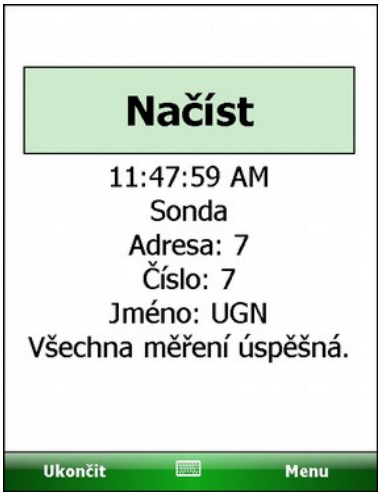
Celé GUI se skládá z následujících dvou základních formulářů: hlavní formulář a konfigurační formulář. Níže jsou vyobrazeny tyto formuláře v hlavních stavech práce. Aplikace má implementovány dva pomocné formuláře.

Obr. 7 – Hlavní obrazovka, pokročilá

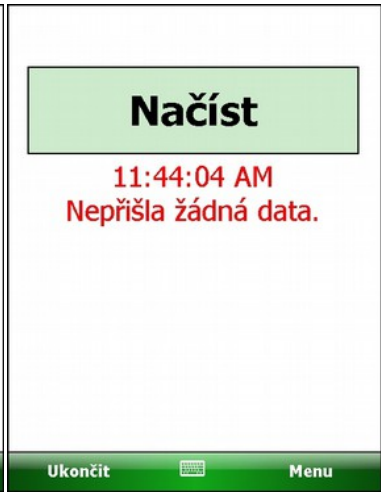
Obr. 8 – Hlavní obrazovka, základní

Na snímcích výše lze vidět, jak se aplikace chová v případě zapnutého či vypnutého pokročilého ovládání. Toto ovládání lze zapnout pomocí konfiguračního souboru a vypnout tamtéž nebo v konfiguračním formuláři.

Obr. 9 – Aplikace při práci se sondou



Obr. 10 – Aplikace po úspěšném měření



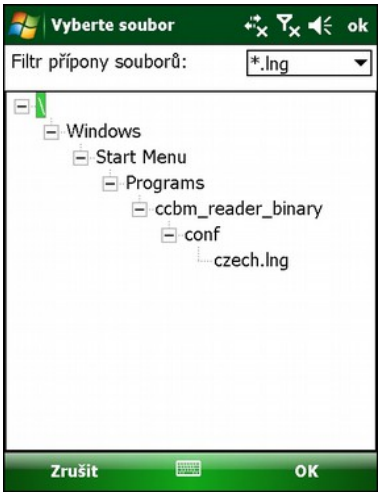
Obr. 11 – Aplikace po neúspěšném měření

Jak lze vidět, aplikace graficky zobrazuje průběh měření. Po každém dokončeném měření se zobrazí zpráva o výsledcích měření. V případě jakéhokoliv neúspěchu je zpráva z důvodu zvýšení přehlednosti zvýrazněna červenou barvou.

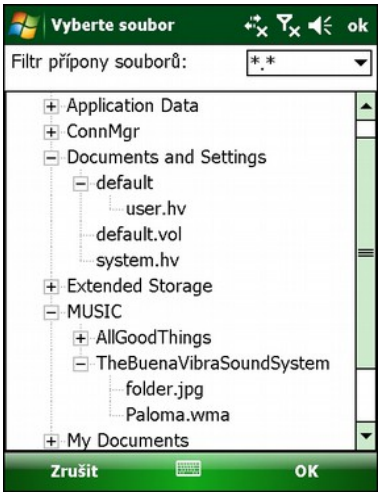
Obr. 12-14 – Konfigurační formulář (jednotlivé karty)

Na obrázcích lze vidět konfigurační formulář. Všechny ovládací prvky jsou přehledně rozděleny do jednotlivých karet. Všechny prvky zobrazují aktuální nastavení aplikace.

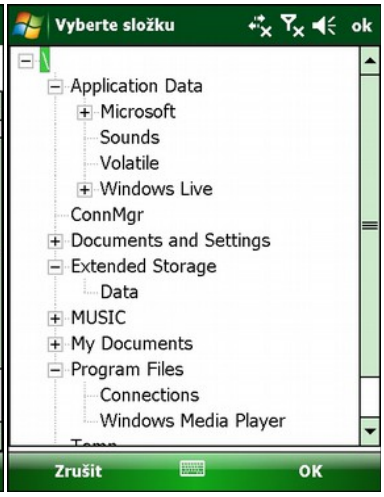
Seznamy s kanály a jejich názvy mají implementovánu synchronizaci zobrazení, takže uživatel nemusí dohledávat odpovídající položky.



Obr. 15 – Výběr souboru překladu (s aktivním filtrem přípony LNG)



Obr. 16 – Výběr souboru překladu (aktivní filtr zobrazuje všechny soubory)



Obr. 17 – Výběr složky pro ukládání dat

Zde jsou vyobrazeny formuláře pro výběr souboru a složky. Tyto formuláře jsem implementoval sám, jak je uvedeno v kapitole 5.2.

6. Detaily řešení

6.1. Ošetření problémů při komunikaci

Při komunikaci po sériovém portu může docházet k řadě různých chyb. V první řadě se jedná o chyby sériového portu samotného a o chyby při přístupu k němu. Jednou z možných chyb je hardwarová závada sériového portu, její výskyt však nelze předpokládat příliš často. Navíc by se pravděpodobně projevila jako výjimka vyvolaná OS při pokusu načíst nebo otevřít sériový port nebo při pokusu přes něj komunikovat. Všechny tyto chyby jsou v programu ošetřeny, jak je popsáno dále, tudíž se touto chybou nemá smysl dále zabývat.

Další chybou týkající se přímo sériového portu může být chyba při pokusu sériový port otevřít. K této chybě může dojít především v případě, kdy se změní jméno portu přiřazené fyzickému zařízení mezi kontrolou, zda název portu existuje (provádí se pouze při startu aplikace a při změně konfigurace) a okamžikem, kdy dojde k pokusu o použití portu (tedy při každém měření). K této chybě může dojít také v případě, že je port již využíván jinou aplikací, která ho má otevřený. Další možností vzniku této chyby je momentální nepřipravenost OS pro operace se sériovým portem (např. z důvodu vytížení CPU procesy s vyšší prioritou). Aby se eliminovaly dočasné chyby vzniklé na základě vytížení, je pokus o otevření portu vždy proveden v případě chyby vícekrát, a to až do maxima pokusů definovaného nastavením aplikace.

Dále mohou vzniknout chyby při komunikaci samotné. Vzhledem k typu komunikace po sériovém portu, kdy není možné ověřit, zda je připojeno nějaké zařízení, nemusí aplikace dostat vůbec žádnou odpověď na jakýkoliv požadavek. Ke stejné chybě může dojít v případě vadné sondy, která není schopna komunikace. Tato chyba je ošetřena samotným návrhem aplikace – každý pokus o čtení odpovědi se totiž provádí pouze do zadaného maxima, a to včetně pokusu o načtení informací o sondě. Aplikace je realizována tak, že se pokusí nejprve načíst kompletní informace o sondě a pokud se jí to ani po maximálním počtu pokusů nepovede, vrací v adrese sondy adresu broadcastu. Je tedy jasné, že se jí nepovedlo žádnou sondu načíst. V takovém případě se aplikace už nepokouší o další měření a je okamžitě zobrazena chybová hláška. Tímto postupem prakticky nedochází k nutnosti čekat za předpokladu, že uživatel nechtěně stiskl tlačítko pro načtení ještě před připojením sondy.

Další chybou při komunikaci může být odpojení sondy v průběhu měření. Tato chyba se projeví neúspěšnými měřeními na všech následujících kanálech, nicméně měření proběhne až do konce a do souboru budou zapsány hodnoty neúspěšných měření. V nejhorším případě (tedy v případě odpojení sondy ihned po přečtení informací) bude čas měření odpovídat: *spWaitMax + measCnt * chCount * rtrMax * (spClearTime + spWaitMax),*
kde chCount je počet měřených kanálů, ostatní hodnoty jsou popsány v části 6.4.2. Při aktuálních výchozích nastaveních aplikace bude tento časový úsek odpovídat přibližně 42s (přesně podle vzorce 41,48 s, nicméně vzorec počítá pouze čekací časy a ne režii aplikace, která je v porovnání s čekacími časy zanedbatelná).

Aktuálně je aplikace nastavena na zápis hodnoty -1 v případě, že se kanál nepovede úspěšně změřit (rozsah naměřených hodnot je 0 – 65 535).

Výše uvedeným způsobem se budou chovat i jednotlivé kanály za předpokladu, že kvůli hardwarové či firmwarové chybě sondy přestanou zcela odpovídat.

K jediným dalším možným chybám patří přijmutí nekorektních zpráv. Zpráva může být buď neúplná nebo nemusí odpovídat zabezpečovací CRC součet. V obou případech je vygenerována výjimka a ve zpracovávání zprávy se nepokračuje, následuje další pokus o přečtení dat. Také kontrola CRC součtu je prováděna pouze v případě korektní délky zprávy.

6.2. Ošetření minimalizace času měření

V aplikaci je implementován algoritmus pro minimalizaci času měření, a to především z důvodu, aby bylo možné provést více měření při jednom požadavku uživatele. Tím, že se provádí několik měření těsně po sobě, je možné při zpracování dat eliminovat různé dočasné chyby a nepřesnosti měření způsobené hardware sondy.

Vzhledem k uvedeným problémům při komunikaci (viz kapitola 5.2) se nabízely dvě možnosti implementace měřícího algoritmu. Tou nejjednodušší bylo vždy čekat dostatečnou dobu, aby byla jistota, že sonda stihne v limitu odpovědět. Takto prováděné měření však bylo značně dlouhé, protože jako bezpečně dlouhý interval se při testování projevil limit 1 s. Aktuální konfigurace sondy má 23 měřených kanálů. Což při základním požadavku provádět více měření najednou a při výchozí hodnotě 5 měření, znamenalo dobu zhruba dvou až dvou a půl minut, protože některé kanály, jak již bylo uvedeno, neodpoví vůbec, i přesto, že je sondě poskytnuta libovolně dlouhá doba.

Takto realizované měření by pro reálnou práci bylo neúnosně dlouhé. Proto jsem přikročil k variantě optimalizačního algoritmu. Tento algoritmus je postaven na smyčce, která čeká vždy po určenou dobu až do maximální doby a mezi jednotlivými dílčími čekáními testuje, zda přišla kompletní data či nikoliv. V případě, že již jsou k dispozici kompletní data, je smyčka přerušena a měřící rutina pokračuje jejich zpracováním. V případě, že ani po dosažení maximální čekací doby data k dispozici nejsou, vygeneruje se výjimka, celý blok zpracování daných dat se přeskakuje a provádí se další pokus o změření dat.

Jako výhodné se také projevilo dát sondě podstatně kratší interval pro odpověď než výše uvedenou 1 s. V případě, že sonda neodpoví cca do 80 – 120 ms, je velmi nízká pravděpodobnost, že ještě vůbec odpoví. Z tohoto důvodu je výhodnější přejít rovnou k dalšímu pokusu o naměření dat, než mnohem déle čekat na data původní. Firmware sondy se v tomto směru jevil spolehlivým alespoň v tom, že i přesto, že nestihl poslat data, byl schopen prakticky vždy přijmout nový požadavek na jejich opětovné naměření.

Dalším prvkem, kterým se dosáhlo značného zkrácení času měření bylo přidání čekání po předchozím pokusu o naměření dat, ať už úspěšném či nikoliv. Zjistil jsem totiž, že sonda vyžaduje určitý čas pro „vyčištění“ komunikace na sériovém portu. Toto může být způsobeno nesprávným návratem z komunikační rutiny či dalšími prováděnými činnostmi po komunikaci

ve firmware sondy. Je také možné, že se jedná o čas potřebný pro přepnutí vstupu kanálového multiplexoru, protože selhání měření se obvykle objevovalo pouze při prvním pokusu o naměření daného kanálu. Díky této poměrně krátké čekací době (při testování se jevilo 40 ms jako dostatečných) se značně snížil počet neúspěšných prvních měření na jednotlivých kanálech.

Za použití celého tohoto algoritmu se mi povedlo snížit čas potřebný pro změření všech 23 kanálů z původních cca 2 – 2,5 minut přibližně na 18 s (s maximem do cca 21 s), tedy přibližně 6x. Tato délka měření se již jeví jako únosná pro nasazení při reálných měřeních.

6.3. Ukládání naměřených dat

Naměřená data jsou ukládána aplikací do souboru CSV. Ten je tvořen tak, aby splňoval specifikace standardních CSV souborů používaných v běžných programech. Především by se mělo jednat o Microsoft Excel a OpenOffice.org Calc. Bohužel existuje poměrně hodně implementací CSV souboru, a to i přesto, že existuje i specifikace tohoto formátu (RFC4180, viz *[9]*). Mnoho firem, zvláště těch větších, tuto specifikaci porušuje a znesnadňuje tak přenositelnost tohoto formátu mezi různými aplikacemi.

Z toho důvodu bylo nutné do aplikace přidat možnost nastavit oddělovač sloupců. Ten bývá nejčastější odlišností u různých aplikací. Obvykle bývá jako oddělovač použita čárka (,) v anglicky mluvících zemích – používá se zde desetinná tečka. Zato v Evropě je obvyklejší používat jako oddělovač středník (;), protože se zde používá desetinná čárka. Záleží pak tedy mnohdy na nainstalované jazykové mutaci Microsoft Windows, Microsoft Office, OpenOffice.org. Bohužel ani za předpokladu, že víme, jaká je instalovaná jazyková mutace, není zaručeno výše uvedené chování. Proto bylo nejjednodušší do programu přidat možnost volby.

Další typickou odlišností jsou textové hodnoty. Obvykle se používá pro uvádění textu dvojitých uvozovek ("). Někdy je však implementováno použití jednoduchých uvozovek ('). V programu Microsoft Excel se uvozovky dokonce nevyužívají vůbec. Vzhledem k tomu, že výstup měl být kompatibilní především s programem Microsoft Excel, a také proto, že jediné textové hodnoty v souboru budou v prvním řádku názvy kanálů, bylo rozhodnuto, že aplikace bude přizpůsobena formátu Microsoftu, a tedy nebudeme využívat uvozovek vůbec. Zvažovala se opět možnost nastavení, avšak aplikace a ukládání dat by tak celkem zbytečně nabývaly na složitosti. Navíc veškerý testovaný software pro zpracování dat si dokázal poradit s použitím textu bez uvozovek.

Všechny datové soubory mají název tvořen adresou, číslem a názvem sondy ve formátu *adresa_cislo_nazev.csv*. Aplikace zapisuje všechna měřená data na konec datových souborů podle identifikace sondy.

V aplikaci jsem také implementoval kontrolu stávajících datových souborů. Při každém ukládání naměřených dat je vytvořena hlavička. V případě existence datového souboru pro danou sondu je ověřována totožnost vygenerované hlavičky a hlavičky v souboru. Jestliže hlavičky neodpovídají, je původní datový soubor přejmenován a označen časovou značkou.

Poté se vytvoří nový datový soubor s hlavičkou odpovídající aktuálním měřeným datům. Tímto postupem je zajištěno, že nemůže dojít k zápisu dat do souboru s nesprávnou hlavičkou.

6.4. Konfigurace programu

6.4.1. Konfigurační modul

Hlavním úkolem konfiguračního modulu je správa konfigurace aplikace. Za tímto účelem je modul vybaven rutinami pro načítání a ukládání konfiguračního souboru, převod textové podoby konfigurace na typy hodnot používané v aplikaci (například číslo, pole čísel atd.), ověření správnosti konkrétní konfigurace a změnu konkrétní konfigurační hodnoty.

Kromě toho má také za úkol udržovat informace o dalších hodnotách, které nejsou v programu nastavitelné, ale jsou využívány na více místech a jejich zadávání a změna na všech místech aplikace by byla zbytečně složitá. Jedná se například o sekvenci konce řádku, nastavení sériového portu (kromě jména, které je uživatelsky nastavitelné), kódování znaků a další.

Konfigurační modul má již při svém načtení ve všech proměnných přednastavené výchozí hodnoty, čímž je zajištěno, že je schopen vytvořit standardní konfiguraci bez jakéhokoliv externího souboru (který by mohl být omylem pozměněn či smazán). Všechny nastavitelné hodnoty lze měnit za běhu programu. Načtení konfiguračních hodnot je umožněno dvěma způsoby, a to načtením z konfiguračního souboru při startu aplikace, nebo metodami pro změnu hodnot za běhu. Tyto metody jsou v aplikaci obsluhovány konfiguračním formulářem. Obě tyto možnosti jsou popsány v následujících kapitolách.

6.4.2. Konfigurační soubor

Konfigurační soubor je základní součástí aplikace. Při každém startu z něj aplikace načítá veškerá nastavení. Za předpokladu, že konfigurační soubor při startu aplikace neexistuje, aplikace nabídne vytvoření nového konfiguračního souboru s výchozími hodnotami.

Konfigurační soubor je uložen standardně s příponou CFG, nicméně jedná se o prostý text v UTF8 formátování. Formát konfiguračního souboru je velice podobný formátu INI souboru, avšak nepodporuje všechny vlastnosti INI souborů (jako například sekce, označení hodnoty jako řetězce a další).

Při tvorbě či editaci tohoto souboru je třeba, aby bylo dodrženo kódování UTF8 a také správné konce řádků specifické pro systém Windows.

Konfigurační soubor umožňuje zadávat pouze jednu hodnotu na každém řádku. Přesný formát je *klíč=hodnota*. Díky tomu je zpracování tohoto souboru programem poměrně snadná a především rychlá záležitost.

O načítání konfiguračního souboru se stará konfigurační modul aplikace. Ten načte celý soubor, rozdělí ho na řádky, a každý řádek na dvě části (konfigurační klíč a konfigurační hodnota). U obou částí se oříznou prázdné znaky (tzv. whitespace) a ověří se, že ani klíč ani hodnota není po tomto oříznutí prázdná. Poté se ověří, jestli hodnota načtená ke konkrétnímu

klíči je korektní konfigurace. Za předpokladu, že ano, hodnota se trvale uloží do nastavení konfiguračního modulu.

V tabulce níže jsou uvedeny všechny konfigurační klíče, jejich popis a platné hodnoty, případná ověření, které konfigurační modul provádí a výchozí hodnoty.

Konfigurační klíč	**Popis**	**Možné hodnoty**	**Výchozí hodnota**
showAdvanced	Určuje, zda budou zobrazeny pokročilé prvky (nabídka ukončení aplikace a nastavení aplikace za pomoci konfiguračního formuláře – viz kapitola 5.4).	0 nebo 1	1
dataDir	Adresář pro ukládání CSV souborů s naměřenými daty, v době startu aplikace musí existovat. V opačném případě se aplikace ptá, zda se má adresář vytvořit. Pokud adresář nebude vytvořen, aplikace nenaběhne.	řetězec	\Extended Storage\Data
langFile	Cesta k souboru s překladem. Tento soubor musí existovat v době startu aplikace. Cesta může být buď absolutní (začínající znakem „\“ – kořenový adresář u Windows Mobile), nebo relativní (cesta vzhledem k adresáři „conf“ umístěném v adresáři aplikace). Pro angličtinu se ponechává hodnota prázdná.	řetězec	prázdný
csvDelim	Jediný znak, oddělující hodnoty v CSV souboru. Implementován z důvodu rozdílnosti specifikací CSV (záleží na použitém software a mnohdy i jeho lokalizaci, zda se používá znak čárky nebo středníku).	1 znak	;
timeFormat	Určuje formát času pro ukládání časové značky do CSV souboru k jednotlivým měřením. Odpovídá specifikaci podle *[10]*.	řetězec	yyyy-MM-dd HH:mm:ss
chNums	Pole čísel kanálů, které se budou načítat ze sondy. Zapisuje se ve formátu {ch1, ch2, ch3,…}.	pole čísel 0-255	

Konfigurační klíč	Popis	Možné hodnoty	Výchozí hodnota
chNames	Pole názvů kanálů, které se budou zapisovat do hlavičky CSV souboru. Pořadí je odpovídající pořadí kanálů. Formát stejný jako u kanálů – {name1, name2, name3,…}.	pole řetězců	
spName	Název sériového portu používaného OS, obvykle COM1 u konkrétního modelu PDA.	COM0-COM255	COM port existující v systému s nejnižším číslem
rtrMax	Maximální počet pokusů o načtení odpovědi na jeden požadavek.	0-65535	3
measCnt	Počet měření uskutečněných na jeden požadavek uživatele.	0-65535	5
spWait	Minimální čas čekání na odpověď sondy, zároveň čas inkrementace v čekací smyčce při čekání na odpověď sondy (v milisekundách).	0-65535	10
spWaitMax	Maximální čas čekání na jednu odpověď (v milisekundách).	0-65535	80
spClearTime	Čas použitý pro „vyčištění“ komunikace na sériovém portu sondy (v milisekundách).	0-65535	40

V položkách chNums a chNames jsou přednastaveny výchozí hodnoty, které zde z důvodů ochrany dat ÚGN neuvádím. Konkrétně se jedná o 23 kanálů a jejich názvy.

6.4.3. Konfigurační formulář

Konfigurační formulář lze zobrazit za pomoci menu v programu, ovšem pouze za předpokladu, že je v konfiguračním souboru povoleno zobrazení pokročilých prvků uživatelského rozhraní. V aktuální implementaci umožňuje konfigurační formulář nastavit všechny hodnoty nastavitelné přes konfigurační soubor. Toto však není nezbytně nutné, protože konfigurační modul není na tomto formuláři nijak závislý. Lze tedy vytvořit i implementaci, ve které by se v konfiguračním formuláři objevovaly pouze důležitější hodnoty nebo například pouze hodnoty použité pro sondu a hodnoty pro celou aplikaci by se nastavovaly pouze přes konfigurační soubor.

6.5. Jazykové verze programu

Na základě požadavku byla aplikace vyvinuta tak, aby umožňovala poměrně jednoduché nasazení jazykových mutací. V aplikaci je obsažen modul pro načítání jazyků ze souboru. Pro snadnou možnost překladu jsou soubory překladu obyčejné textové soubory ve formátu podobném konfiguračnímu souboru programu. Překlad se zde zadává stejně, tedy *klíč=hodnota*. Také pro zaručení maximální kompatibility s co nejvíce jazyky je kódování jazykového souboru nastaveno na celosvětový formát UTF8, který obsahuje prakticky všechny používané znaky.

Díky tomu je možné vytvořit soubor překladu i bez jakýchkoliv speciálních nástrojů. Lze ho vytvořit libovolným textovým editorem, jedinou podmínkou je, aby zvládal uložení souboru v kódování UTF8 a se správnými konci řádku, které jsou specifické pro systém Windows.

Není nutné, aby v souboru překladu byly uvedeny všechny popisky a hlášky. Jazykový modul a celá aplikace jsou koncipovány tak, aby byl použit kterýkoliv překlad, který je nalezen v zadaném souboru, pokud odpovídá překladu potřebnému v používané části aplikace.

7. Stav projektu a plánovaný další vývoj

7.1. Stav projektu

První verze aplikace byla vyvinuta, avšak vzhledem k značnému množství problémů způsobených nesprávnou, časově nepřesnou a někdy nepředvídatelnou komunikací sondy nebyla nikdy nasazena do provozu. Aktuálně je vyvinuta druhá generace této aplikace. Ta již obsahuje kompletní ošetření chyb komunikace se sondou a pokročilý konfigurační formulář, který aktuálně obsahuje všechna nastavení konfiguračního modulu. Aplikace také má implementován algoritmus pro minimalizaci času potřebného pro měření, díky němuž se dosáhlo zhruba šestinásobného zrychlení oproti první verzi (jsou-li správně nastaveny časy čekání na sondu).

Tato verze je již nasazena do produkčního prostředí, je testována a v nejbližší době bude nasazena pro použití při reálných měřeních.

Vzhledem k plánovaným rozšířením je ale předpokládáno poměrně brzké nahrazení této verze verzí následující.

7.2. Plánovaný vývoj

Aplikace již nyní má plánovaný další vývoj. Především se jedná o záznam ladících informací o průběhu měření, díky kterým bude možné vyladit ještě přesněji nastavované parametry sondy. Tyto záznamy by měla být aplikace schopna samostatně vyhodnotit a nabídnout optimální časy čekání. Všechna doporučená nastavení budou i nadále upravitelná uživatelem, aby bylo v případě potřeby možné sondu přenastavit. Do aplikace bude také implementována funkce pro cílené testování rychlosti měření, aby bylo možné vyhodnotit optimálnost nastavení.

Další funkčností by měla být možnost měření různými typy sond. Aktuální aplikace totiž standardně umožňuje pouze jedinou konfiguraci čekacích časů a kanálů sondy. Do budoucna je plánováno rozdělení konfigurace na globální nastavení (tedy název sériového portu, počet měření, soubor překladu atd.) a nastavení typické pro sondu (tedy kanály, jejich názvy, čas čekání, počet pokusů atd.). Tyto konfigurace typické pro jednotlivé sondy by se měly být schopny načítat automaticky, podle toho, která sonda bude připojena – rozpoznání bude založeno na adrese, čísle nebo názvu sondy, či jejich kombinaci, o tom není zatím přesně rozhodnuto.

Z důvodů zabezpečení naměřených dat bude v příští verzi implementováno ukládání všech dat do dvou různých míst. Obě místa budou samozřejmě záležet na výběru uživatele, nicméně měla by se volit dvě různá fyzická média, aby nemohlo dojít ke ztrátě dat poškozením jednoho či druhého média. Vzhledem k aktuálnímu vybavení PDA by se mělo jednat o vnitřní úložiště PDA (4GB iNAND trvalá paměť) a o SD kartu.

Také je plánováno rozšíření této aplikace o možnost automatizovaného odesílání naměřených dat na server, který by tato data byl schopen shromažďovat z několika PDA,

několika sond a možná i několika různých umístění. Měl by také být schopen tato data archivovat a ukládat v celistvé podobě tak, aby pro koncového uživatele nebylo patrné, zda byla nebo nebyla měřena různými PDA či pracovníky dolů a byla možnost okamžitě pracovat s kompletními daty.

8. Testování a hodnocení

Testování aplikace probíhalo na dvou úrovních, a to na úrovni uživatelské a vývojářské.

Uživatelské testování aplikace se uskutečňovalo za pomoci pracovníků ÚGN, kteří jsou také potencionálními uživateli aplikace. Toto testování mělo především ověřit uživatelskou přívětivost a spolehlivost v běžném (simulovaném) provozu. Simulace provozu spočívala v práci se sondou, tedy v jejím opakovaném měření. V průběhu simulace byly také ověřeny případy hardwarového selhání sondy nebo jejích součástí.

Vývojářské testování aplikace probíhalo ve dvou fázích, a to nejprve na emulátoru (viz kapitola 1.5) a poté na reálném hardware. V obou případech byl použit testovací modul odpovídající hardwarovému i firmwarovému vybavení sond, pouze nebyl osazen modul kanálového multiplexoru, tedy na všech kanálech byly měřeny stejné hodnoty. Při tomto testování byla ověřována také ovladatelnost aplikace, ale především její stabilita. Aplikace byla testována pro případy nesprávné konfigurace, nekorektních konfiguračních souborů, nekorektních souborů překladu a také pro případ úplné ztráty připojení sondy v průběhu komunikace při měření. V neposlední řadě byly testovány různé násilné uživatelské zásahy v době běhu různých operací, generována nesprávná nastavení a odpovědi. Dále byly do zdrojových kódu přidávány rutiny pro generování chyb a ověřovány reakce jejich ošetření. V průběhu tohoto testování bylo provedeno několik optimalizací aplikace tak, aby jakákoliv chyba, ať už vyvolaná selháním aplikace, hardware nebo uživatele měla co nejmenší dosah na spolehlivost a na korektní běh aplikace, či aby se alespoň minimalizoval dopad způsobených „škod“.

V průběhu testování se postupně dle požadavků pracovníků ÚGN upravovaly detaily, především v oblasti uživatelského rozhraní a korektních a srozumitelných popisků. Ovladatelnost aplikace, rychlost měření sondy i spolehlivost a odolnost aplikace vůči chybám byly bez výhrad splněny.

9. Závěr

Navrhl jsem kompletní software pro záznam dat přenášených z napěťových sond – CCBM Reader. Tento software je navržen s ohledem na funkčnost, hardwarové i softwarové schopnosti a možnosti využitého kapesního počítače Nautiz X7. Software se při testování jevil jako spolehlivý, stabilní, přehledný a uživatelsky přívětivý. Navržený software bez výhrad splňuje všechny požadavky zadavatelské organizace Ústavu geoniky Akademie věd ČR, v.v.i.

Konkrétně bylo díky správnému ošetření chyb vznikajících při komunikaci a algoritmu pro minimalizaci času dosaženo přibližně 6x rcyhlejšího měření než u první verze této aplikace a 8x rychlejšího měření než u původně vyvinuté aplikace určené pro počítače s OS Windows. Moje aplikace je také mnohem stabilnější. Další výhodou mojí aplikace je zvolené průběžné ukládání, protože i v případě jakékoliv chyby – ať už softwarové či hardwarové – jsou všechna dosud naměřená data správně uložena. Původní aplikace pro OS Windows ukládala veškerá data až při ukončení celé aplikace, čímž značně zatěžovala paměť RAM, které je v PDA poměrně nedostatek a také vždy došlo ke ztrátě dat v případě jakékoliv chyby.

V současné době jediným známým nedostatkem je nutnost odhadnout a manuálně otestovat časy pro komunikaci se sondou. Tyto časy aktuálně není možné nechat spočítat či vyhodnotit aplikací. Odstranění tohoto nedostatku by vyžadovalo značně netriviální řešení. Tato funkčnost by měla být implementována spolu s ostatními novými funkcemi v příští hlavní verzi aplikace.

Jediným dalším nedostatkem by se mohlo stát, že aplikace je prakticky nepřenositelná na novější zařízení, která již nemají OS Windows Mobile. Jak již bylo uvedeno, aktuální generace sondy vyžaduje napájení z externího zdroje. Tato koncepce by měla být zachována i pro budoucí generace napěťových sond. Z tohoto důvodu je nutné pro každé PDA ještě vytvořit napájecí modul, který bude umístěn a upevněn ve spodní části PDA. Toto je jedním z hlavních důvodů, proč uvedený nedostatek může být zanedbán, protože přechod na novější platformu či jiný hardware není plánován.

Literatura

[1] WIKIPEDIA, the free encyclopedia. Newline
URL: <http://en.wikipedia.org/wiki/Newline> |cit. 2013-04-15|.

[2] NAKAMURA, N., OHKUBO, R., OBARA, Y., KANG, S. S., SUGAWARA, K. AND KANEKO, K. ROCK STRESS MEASUREMENT FOR LIMESTONE OPEN PIT MINE. In Proceeding of 5th International Symposium on field Measurements in Geomechanics. Singapore, Balkema, Rotterdam, 1999, pages 375-380.

[3] KALÁB T., STAŠ L., RNDR., CSC., KNEJZLÍK J., ING., CSC. Popis měřicích sond CCBM a adaptace jejich standardního provedení na podmínky tonalitu pro použití ve štole Josef v rozrážce SP-47 pro účely projektu FR–TI–3/325
Ústav geoniky AV ČR, v.v.i., prosinec 2011, 12 stran

[4] TEDIA. AIBus 2, specifikace komunikačního protokou
Dostupné z: <http://www.tedia.cz/download/files/aibus2.pdf>.

[5] Windows CE installation cabinet (.CAB) file format
<http://www.cabextract.org.uk/wince_cab_format/> |cit 2013-04-15|.

[6] MICROSOFT. Creating an .inf File
URL: <http://msdn.microsoft.com/en-us/library/ms924764.aspx> |cit. 2013-04-15|.

[7] MICROSOFT. CAB Wizard (Windows CE 5.0)
<http://msdn.microsoft.com/en-us/library/aa448616.aspx> |cit 2013-04-15|.

[8] MICROSOFT. .NET Compact Framework
<http://msdn.microsoft.com/en-us/library/f44bbwa1%28v=vs.90%29.aspx> |cit 2013-04-15|

[9] MICROSOFT. Custom Date and Time Format Strings
URL: <http://msdn.microsoft.com/en-us/library/8kb3ddd4%28v=vs.90%29.aspx> |cit. 2013-04-15|.

[10] SHAFRANOVICH, Y. RFC 4180 – Common Format and MIME Type for Comma - Separated Values (CSV) Files
URL: <http://tools.ietf.org/html/rfc4180> |cit. 2013-04-15|.

Přílohy

I. CD se zdrojovými kódy aplikace (neveřejná část)